TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Roberto Airoldi

Design and Implementation of Software Defined Radios on a Homogeneous Multi-Processor Architecture

Julkaisu 1136 • Publication 1136

Tampere 2013

Tampereen teknillinen yliopisto. Julkaisu 1136
Tampere University of Technology. Publication 1136

Roberto Airoldi

Design and Implementation of Software Defined Radios on a Homogeneous Multi-Processor Architecture

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 30th of May 2013, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2013

Supervisor:

Professor Jari Nurmi
Department of Electronics and Communications Engineering
Tampere University of Technology
Tampere, Finland

Pre-examiners:

Professor Donatella Sciuto
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Milano, Italy

Professor Michael Hübner
Lehrstuhl für Eingebettete Systeme der Informationstechnik (ESIT)
Ruhr-Universität Bochum
Bochum, Germany

Opponents:

Professor Michael Hübner
Lehrstuhl für Eingebettete Systeme der Informationstechnik (ESIT)
Ruhr-Universität Bochum
Bochum, Germany

Professor Hannu Heusala
Department of Electrical and Information Engineering
University of Oulu
Oulu, Finland

ISBN 978-952-15-3078-4 (printed)
ISBN 978-952-15-3096-8 (PDF)
ISSN 1459-2045

ABSTRACT

In the wireless communications domain, multi-mode and multi-standard platforms are becoming increasingly the central focus of system architects. In fact, mobile terminal users require more and more mobility and throughput, pushing towards a fully integrated radio system able to support different communication protocols running concurrently on the platform. A new concept of radio system was introduced to meet the users' expectations. Flexible radio platforms have became an indispensable requirement to meet the expectations of the users today and in the future.

This thesis deals with issues related to the design of flexible radio platforms. In particular, the flexibility of the radio system is achieved through the concept of software defined radios (SDRs). The research work focuses on the utilization of homogeneous multi-processor (MP) architectures as a feasible way to efficiently implement SDR platforms. In fact, platforms based on MP architectures are able to deliver high performance together with a high degree of flexibility. Moreover, homogeneous MP platforms are able to reduce design and verification costs as well as provide a high scalability in terms of software and hardware. However, homogeneous MP architectures provide less computational efficiency when compared to heterogeneous solutions.

This thesis can be divided into two parts: the first part is related to the implementation of a reference platform while the second part of the thesis introduces the design and implementation of flexible, high performance, power and energy efficient algorithms for wireless communications. The proposed reference platform, Ninesilica, is a homogeneous MP architecture composed of a 3x3 mesh of processing nodes (PNs), interconnected by a hierarchical Network-on-Chip (NoC). Each PN hosts as Processing Element (PE) a processor core. To improve the computational efficiency of the platform, different power and energy saving techniques have been investigated. In the design, implementation and mapping of the algorithms, the following constraints

were considered: energy and power efficiency, high scalability of the platform, portability of the solutions across similar platforms, and parallelization efficiency.

Ninesilica architecture together with the proposed algorithm implementations showed that homogeneous MP architectures are highly scalable platforms, both in terms of hardware and software. Furthermore, Ninesilica architecture demonstrated that homogeneous MPs are able to achieve high parallelization efficiency as well as high energy and power savings, meeting the requirements of SDRs as well as enabling cognitive radios.

Ninesilica can be utilized as a stand-alone block or as an elementary building block to realize clustered many-core architectures. Moreover, the obtained results, in terms of parallelization efficiency as well as power and energy efficiency are independent of the type of PE utilized, ensuring the portability of the results to similar architectures based on a different type of processing element.

PREFACE

The work presented here was carried out in the period 2009-2013 in the Department of Computer Systems and in the Department of Electronics and Communications Engineering (during the year 2013) at Tampere University of Technology, Tampere, Finland.

I would like to acknowledge Prof. Jari Nurmi, who made possible the accomplishment of this research work. He welcomed me into his research group, always believed in my research ideas and supervised this thesis work.

I also would like to acknowledge the pre-examiners Prof. Donatella Sciuto and Prof. Michael Hübner, who provided valuable comments to improve this thesis. Moreover, I would like to take the opportunity to thank senior researchers of the working group, who guided my first steps as a research scientist. In particular, Dr. Fabio Garzia has been a truly inspiring instructor and helped me to find my own way in the research world. I also acknowledge Dr. Fabio Campi from Bologna, who has always showed interest in my research work and provided me with useful feedback whenever I asked.

In giving me the opportunity I would like to express all my gratitude to the co-authors of the published research works that form the basis of this thesis. I am particularly grateful to Prof. Alexander Wyglinski, who guided my first steps in the field of cognitive radios. My colleagues at Tampere University of Technology made my work a pleasant experience, both from the work and human relationship points of view.

The work was financially supported by the *Graduate School in Electronics, Telecommunications and Automation* (GETA) and *Tampere University of Technology*. Research grants were received from the *Tuula ja Yrjö Neuvo Foundation*, the *Nokia Foundation*, the *Ulla Tuominen Foundation* and the *Tekniikan Edistämissätiö*, which are all gratefully acknowledged.

Finally, I would like to thank all the people who have been close to me during this

period. In particular, I am very thankful to Stefano Biancullo, Srinivas Chirumari, Francescantonio Della Rosa and Antonietta Pepe for their friendship and the nice moments shared together. My last acknowledgement goes to my family, which has always supported my decisions and pushed me to do my best in my professional as well as private life.

8th May 2013

Roberto Airoldi

TABLE OF CONTENTS

LIST OF PUBLICATIONS

This thesis is mainly based on the following publications. In the manuscript the publications are referred to as [P#]. The publications are appended at the end of the thesis.

[P1] Airoldi, R.; Garzia, F.; Ahonen, T.; Milojevic, D.; Nurmi J., "Implementation of W-CDMA Cell Search on a FPGA based Multi-Processor System-on-Chip with Power Management", *Proceedings of the IX International Symposium on Systems, Architectures, MOdeling and Simulation (SAMOS IX)*, Springer Verlag (LNCS series), 20-23 July 2009, Samos, Greece, pp. 88-97.

[P2] Airoldi, R.; Garzia, F.; Nurmi J., "FFT Algorithms Evaluation on a Homogeneous Multi-processor System-on-Chip", *Proceedings of the 2010 International Conference on Parallel Processing (ICPP) Workshops*, IEEE, 13-16 September 2010, San Diego, CA, USA, pp. 58-64.

[P3] Airoldi, R.; Ahonen, T.; Garzia, F.; Milojevic, D.; Nurmi J., "Implementation of W-CDMA Cell Search on a Highly Parallel and Scalable MPSoC", *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, Springer US, Vol. 64, Issue 1, 2011, pp. 137-148.

[P4] Airoldi, R.; Garzia, F.; Anjum, O.; Nurmi J., "Homogeneous MPSoC as Baseband Signal Processing Engine for OFDM Systems", *Proceedings of the 2010 International Symposium on System-on-Chip (SOC '10)*, IEEE, 28-30 September 2010, Tampere, Finland, pp. 26-30.

[P5] Airoldi, R.; Anjum, O.; Garzia, F.; Wyglinski, A.M.; Nurmi J., "Energy-Efficient Fast Fourier Transforms for Cognitive Radio Systems", *IEEE Micro Magazine*, IEEE, Vol. 30, Issue 6, 2010, pp. 66-76.

[P6] Airoldi, R.; Garzia, F.; Nurmi J., "Improving Reconfigurable Hardware Energy Efficiency and Robustness via DVFS-Scaled Homogeneous MP-SoC", *Proceedings of the 2011 International Parallel & Distributed Processing Symposium Workshops*, IEEE, 16-17 May 2011, Anchorage, AK, USA, pp. 281-284.

[P7] Airoldi, R.; Garzia, F.; Nurmi J., "Efficient FFT Pruning Algorithm for Non-Contiguous OFDM Systems", *Proceedings of the 2011 Conference on Design and Architectures for Signal and Image Processing (DASIP 2011)*, IEEE, 2-4 November 2011, Tampere, Finland, pp. 1-6.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC	Analog-to-Digital Converter
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction-Set Processor
CGRA	Coarse-Grain Reconfigurable Array
CG	Clock Gating
CN	Computational Node
CR	Cognitive Radios
DMA	Direct Memory Access
DAC	Digital-to-Analog Converter
D-OFDM	Discontiguous Orthogonal Frequency Division Multiplexing
DSA	Dynamic Spectrum Access
DSP	Digital Signal Processor
DVFS	Dynamic Voltage Frequency Scaling
eFPGA	embedded Field-Programmable Gate Array
FFT	Fast Fourier Transform
FIR	Finite Impulse Response

FPGA	Field-Programmable Gate Array
GOPS	Giga Operation Per Second
HDL	Hardware Description Language
HW	Hardware
IC	Integrated Circuit
IP	Intellectual Property
LTE	Long Term Evolution
LUT	Look-Up-Table
MP	Multi-Processor
MPI	Message Passing Interface
MPSoC	Multi-Processor System-on-Chip
NC-OFDM	Non-Contiguous Orthogonal Frequency Division Multiplexing
NI	Network Interface
NoC	Network-on-Chip
OFDM	Orthogonal Frequency Division Multiplexing
PE	Processing Element
PN	Processing Node
PVT	Process Voltage Temperature
QoS	Quality of Service
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RR	Reconfigurable Radio
SCC	Scalable Communication Core

SCR	Software Controlled Radio
SDR	Software Defined Radio
SIMD	Single-Instruction Multiple-Data
SNR	Signal-to-Noise Ratio
SoC	System-on-Chip
UDSM	Ultra-Deep SubMicron
VHDL	VHSIC Hardware Description Language
VLIW	Very Long Instruction Word
VLSI	Very Large Scale Integration
W-CDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Networks

1. INTRODUCTION

The market for wireless communication systems has been one of the most profitable markets in the microelectronics industry for the past two decades [13]. The continuous technological advance of wireless communications, and the consequent increase of the number of available communication protocols, requires the design of complex Systems-on-Chip (SoCs) for the implementation of multi-standard transceivers [3]. However, the implementation of multi-standard platforms introduces new challenges and thus new design approaches are required. In fact, the design of multi-standard transceivers cannot be based on a simple collage of single-standard platforms: energy and power consumption, design and verification costs, as well as silicon area utilization, will not meet the design's constraints. More integrated and flexible solutions have to be explored.

Flexible platforms, able to implement and adapt to any communication system, have been the target of many research groups, both in academia and industry. A flexible platform is able to adapt its functionality on the basis of the users' requirements, enabling the utilization of different wireless communication systems on the same hardware (HW). Moreover, a higher degree of flexibility helps to keep design costs under control because a single architecture can be used for a wider set of applications. In addition to flexibility, other important factors for the implementation of transceivers are the computational power and the computational efficiency. Indeed, the latest communication systems require more and more computational power to provide higher data-rates to the users within a tight power budget.

To achieve high performances, system designers have to rely on the latest silicon technology nodes. In fact, the high density provided by submicron technologies enables the implementation of highly complex SoCs, leading to high performances in terms of giga operations per second per squared millimeter ($GOPS/mm^2$). On the other hand, the latest technology nodes are characterised by high power densities [26]

which become a limiting factor for the design of SoCs. In fact, high power densities on the chip leads to hard and soft errors (e.g. electro-migrations and IR Drops) which have a negative impact on the yield. Therefore, power consumption is not any more just a design parameter important for mobile applications, but has become one of the main constraints in the design of modern SoCs. Moreover, the utilization of ultra-deep submicron (UDSM) technologies requires large financial investments to cover the high costs related to design and verification of complex SoCs as well as the intrinsic higher costs of the silicon real estate. To counterbalance the financial investment, the design effort of Very Large Scale Integration (VLSI) systems must be shared across different families of products and, if possible, across different application domains [12]. To enable the re-utilization of platforms, a high degree of flexibility from the architectural point of view is required.

From the flexibility and the high performance points of view, platforms based on multi-processor architectures seem appealing because of their high level of programmability and computational power. On the other hand, such architectural choices introduce other issues that designers have to deal with [42]. One of the toughest design challenges related to the utilization of multi-processor architectures resides in the programming model and in particular, in the software concurrency. In addition to software issues, other problems arise: synchronisation between computational nodes, task scheduling, data distribution and power management are just a few examples of the issues that have to be faced when designing multi-processor architectures. Finally, the implementation dilemma: heterogeneous or homogeneous multi-processor architecture. Heterogeneous solutions provide a better fit to the application domain while the strength of homogeneous solutions is in the reduced design and verification costs as well as scalability.

Heterogeneous platforms are today the most common architectural solution since they are able to deliver high performance with high efficiency, measured for example in giga operations per second per milliWatts (GOPS/mW). In fact, each processing node of the architecture is optimized to perform a predefined set of tasks. On the other hand, the cost related to the design and verification of such systems dramatically increases with technology scaling. Moreover, at the application level, software engineers have to design software tool-chains able to work with different target architectures, programming languages, compilers and tools.

For these reasons, homogeneous architectures, designed on the base of a copy-and-

paste approach and an extensive utilization of the intellectual properties (IP) reuse concept [47], become appealing because of their native ability to keep design, and especially verification, costs under control. Moreover, the software layer becomes simplified as well. As a drawback, consisting of only a single type of processing element reduces the system efficiency and performance. In other words, the computational node cannot be customized for a single specific task. However, future technology nodes would allow the integration of more and more processing elements, thus mitigating the performance gap.

Looking at the target application domain, new challenges for the development of algorithms are introduced by the utilization of multi-processor platforms. The design and implementation of algorithms must now consider the parallelism exposed by the platforms and provide efficient solutions that lead to a full exploitation of the parallel resources made available by the architecture.

1.1 Objective and scope of the research

The aim of this research work was the implementation and the evaluation of software defined radio (SDR) applications on a reference homogeneous multi-processor architecture. The goal was to define a powerful and flexible enough architecture, which is able to cope with the requirements of the physical layer signal processing of wireless communication systems. In addition to flexibility and computational power, other design parameters were also considered (e.g. power and energy efficiency). Energy and power consumption are indeed crucial when dealing with applications that target battery operated devices. In fact, power and energy have strictly fixed budgets and both software and hardware layers must be as optimized as possible and efficiently co-operate with each other.

1.2 Main results

The main results achieved in this research work are: 1) the definition of a reference homogeneous multi-processor platform (Ninesilica cluster), and 2) the design and implementation of novel algorithms enabling efficient mapping of wireless communications on the proposed reference platform. The proposed platform together with

the design and implementation of the algorithms showed that the utilization of homogeneous architectures for the implementation of baseband signal processing meets all the requirements set by the target applications, with high energy and power efficiency, allowing a high degree of scalability in terms of software and hardware.

1.3 Thesis outline

This manuscript is organized as follows: Chapter 2 introduces the challenges related to the implementation of future wireless communication systems, giving an introduction to SDRs and Cognitive Radios (CRs); Chapter 3 analyses state-of-the-art platforms for the implementation of SDR base-band receivers, discussing different architectural solutions and their advantages and disadvantages; then the design of a reference homogeneous multi-processor architecture for the implementation of SDR algorithms is introduced in Chapter 4; Chapter 5 focuses on the design and implementation of SDR algorithms on the proposed reference platform; Chapter 6 summarizes the publications and the author's contributions to the published works, which form the base of this thesis; finally, concluding remarks, open issues and future directions are discussed in Chapter 7. Appendix A contains the peer-reviewed publications [P1-7] on which the thesis is mainly based.

2. REQUIREMENTS OF FUTURE WIRELESS COMMUNICATIONS

This chapter analyzes the requirements for the design of platforms for the implementation of future wireless communications. Because of the shift towards multi-mode and multi-standard transceivers, high importance is today given to flexible radios [37]; a high degree of flexibility in the transceiver allows the system architects to adapt the platform to different communication protocols, to keep the design and development costs under control and to shorten the time to market as well. In fact, from a production point of view, the utilization of flexible radio platforms enables the re-utilization of the same radio architecture for different application domains, which allows the sharing of the design and implementation costs of the platform across different application domains and products.

2.1 Software defined radio: the required flexibility

The flexibility of the radio systems is a very wide concept and can be applied to all of the OSI layers of the communication system [46]. Therefore, some restrictions and classifications are required to correctly focus on the research area covered by this thesis. This thesis work focuses only on the physical layer aspects of wireless communications, and therefore it considers the radio flexibility from the digital baseband point of view.

To implement a flexible radio, different approaches can be followed. These approaches can be classified on the basis of the achieved flexibility and architectural complexity. Fig. 1 presents different flexible radio solutions on a flexibility - architectural complexity plane. As shown in the figure, a pure software radio, where all the radio functionalities are executed in software, offers the highest flexibility, virtually an infinite level of flexibility, but with the downside of an overly high architectural complexity. Conversely, software controlled radios (SCRs) are simpler

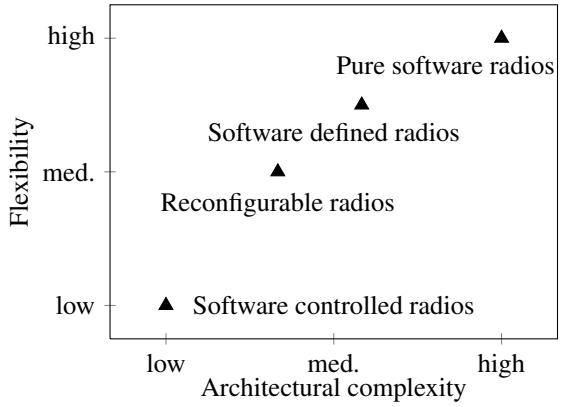


Fig. 1. *Flexibility - architectural complexity map for different flexible radio approaches.*

platforms and provide low levels of flexibility. In fact, the software layer is only utilized for the selection and control of a few parameters of the communication system. In between these two opposite approaches, other solutions can be identified. In particular, software-centric platforms are gaining more popularity due to the technology advancement in the digital signal processing field.

SDR forum [1] defines software defined radio as a radio where the physical layer functionalities are performed entirely (or partially) in software. The following subsections give a description of the classification of software radios followed by this research work.

Software controlled radios

In software controlled radios, the software layer has the role of controlling the behaviour of the radio system. Fig. 2 presents an example schematic of a software controlled radio. The core functionalities of the radio are performed in dedicated hardware blocks and the software selects which chain has to be executed. From a flexibility point of view, this approach is quite limited and does not scale with respect to system performance. On the other hand, the system complexity is low.

Reconfigurable radios

Reconfigurable radios (RRs) are a step ahead of SCRs in achievable flexibility. This higher level of flexibility is obtained by the introduction of elements of reconfigurability in their radio chains. For example, constellation selection or Fast Fourier Transform (FFT) size for Orthogonal Frequency Division Multiplexing (OFDM) demodulation can be adjusted by reconfiguring hardware blocks. RRs offer a medium level of flexibility and a medium level of complexity, enabling efficient implementations of multi-mode radios. However, the achieved flexibility level is not high enough to support multi-standard radio systems.

Software defined radios

A further step towards a flexible radio is achieved via the implementation of a software defined radio, which allows a high degree of flexibility with the drawback of a medium high architectural complexity. In a SDR platform most of the signal processing is done in processor cores (generally Digital Signal Processors - DSPs) and therefore the high level of flexibility introduced by the software allows the implementation of multi-mode, multi-standard transceivers. Limitations in terms of architectural complexity are generally related to the real-time requirements and the high computational workload of the algorithms. However, the scaling of silicon technology makes available an increasing number of transistors for the implementation of more computationally powerful architectures, making SDR a concrete approach.

Pure software radios

A pure software radio implements all the functional kernels of the physical layer at software level. Therefore, the schematic view of the radio platform looks quite simple, as Fig. 3 shows. However, the architectural complexity of the system exceeds the limits of today's technology. For example, Digital-to-Analog Converters (DACs) and Analog-to-Digital Converters (ADCs) are today a limiting factor due to their limited achievable bandwidth. However, even if semiconductor technology offered architectural solutions able to meet the requirements of pure software radios, there might be no reason to implement it, making the pure software radio an ideal model and SDR its real implementation [29].

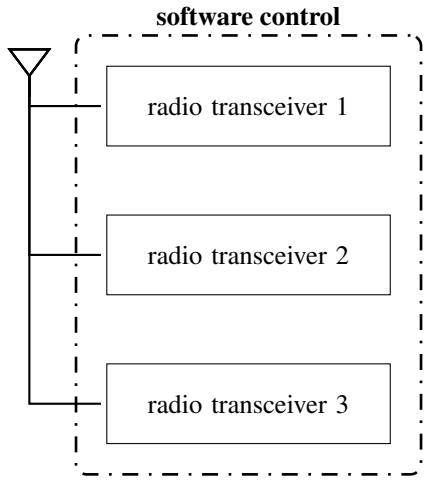


Fig. 2. *Simplified schematic view of a software controlled radio receiver.*

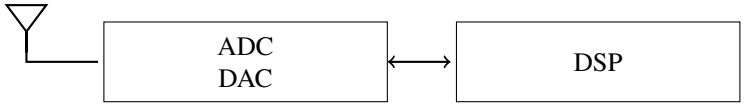


Fig. 3. *Schematic view of a pure software radio receiver.*

2.2 Towards cognitive radios

With the increasing demand for additional bandwidth due to existing and new services, both spectrum policy makers and communication technologists are seeking solutions for resolving the spectrum scarcity issue. However, a recent study [20] has shown that this scarcity of available wireless transmission bandwidth is the result of the spectrum not being efficiently utilized across frequency, space, and time. Consequently, researchers from both industry and academia have commenced work on developing new paradigms for mitigating the spectrum scarcity issue. One proposed paradigm for efficiently utilizing spectrum is dynamic spectrum access (DSA) [58]. In order to realize DSA communications, highly reconfigurable wireless platforms possessing a high level of spectral awareness are needed. Using flexible and efficient baseband processing that is available through SDR technology, *cognitive radio* [39] can be devised for enabling DSA communications.

2.3 Architectural requirements

This section summarizes the architectural requirements for the design of a radio system, which is able to implement on the same platform multi-mode multi-standard transceivers. The support of future wireless communication systems and cognitive radios are central design specifications. A radio system should provide:

- *Flexibility*: flexibility is one of the main requirements in order to support multi-mode transmission, multi-standard transceivers and to enable cognitive radios.
- *High computational power*: high computational power ensures that the strict real-time requirements of the latest wireless communication systems are met. In addition, the computational power provided by the architecture should be scalable to be able to meet the requirements of future standards, without the need to completely redesign the platform.
- *Power efficiency*: radio systems work with a tight power budget, e.g. just $200mW$ are allocated for the receiver side in 3gpp Long Term Evolution (LTE) system [50], and therefore power efficiency is one of the main design parameter for radio platforms. Moreover, to enable the utilization of the latest technology nodes system architects must consider the high impact of power consumption on the yield and reliability of the platform [41].

On the basis of these requirements it is possible to identify architectural solutions that might be more suitable than traditional design approaches. A design solution that has been widely adopted for the implementation of SDR is based on multi-processor architecture, or more in general on Multi-Processor Systems-on-Chip (MPSoCs) [34]. In fact, MPSoCs, based on processor cores, offer high computational power and a high degree of flexibility. As a drawback, new challenges are arising in different domains, e.g. programming model and power consumption. However, an efficient synergy between platform development and algorithm design and implementation can mitigate these issues.

3. PLATFORMS FOR SOFTWARE DEFINED RADIOS

This chapter analyses the architectural approaches utilized for the realization of SDR platforms. In particular, the chapter focuses on digital base-band processing. Therefore, of the whole SDR platform, which also generally includes processing for the front-end as well as for higher layers of the communication protocol, only the base-band engine is considered. Particular emphasis is then given to existing reconfigurable hardware and multi-processor solutions. Finally, limitations of such design approaches are also underlined and discussed.

3.1 Reconfigurable architectures

The world of reconfigurable systems is quite scattered and includes many different levels of reconfigurability. Different flavours of parallelism are tackled by different levels of granularity. However, for the digital base-band signal processing a coarse level of reconfigurability is generally required and therefore most of the proposed architectures are coarse grained reconfigurable architecture (CGRA) [10, 21, 25, 38, 52].

3.1.1 Reconfigurable Radio architecture

The reconfigurable radio proposed by Zhang et al. in [57] is a heterogeneous multi-processor platform. The architecture is composed by a mesh of different processing elements interconnected by a Network-on-Chip (NoC). The architecture includes general purpose processors, embedded Field Programmable Gate Arrays (eFPGAs), Application-Specific Integrated Circuits (ASICs) and reconfigurable HW blocks. The digital base-band functionalities are mapped on the reconfigurable hardware blocks, which are based on Montium architecture [25]. Montium was designed to target 16-bit DSP algorithms. Montium architecture looks like a Very Long Instruction Word

(VLIW) processor. However, its control structure is quite different from that of processor cores. In fact, instructions are statically scheduled so that the activity of the decoding unit is reduced to the minimum [55]. The proposed architecture is flexible enough to support the implementation of OFDM-based cognitive radios.

3.1.2 Scalable Communication Core

The Scalable Communication Core (SCC) was designed by Intel as a flexible platform to support simultaneously a combination of WiFi, WiMax and broadcasting digital TV communication systems [16]. The digital base-band processing of the SCC is based on reconfigurable processing elements interconnected to each other via a NoC [17]. The processing elements support multi-threading in order to enable the concurrent processing of multiple communication protocols. The heterogeneous mix of processing elements made available by the SCC platform allows the optimization of each single IP to a specific set of algorithms required by the target applications. As an example, SCC provides reconfigurable elements that specifically target forward error correction algorithms. Moreover, specific data streaming processors have been designed in order to provide support to a wide range of fixed point mathematical operations.

The research efforts made by Intel during the design and evaluation of the SCC are summarized in [16]. Particular emphasis is given to the achieved results and to the open issues that have not been solved. In particular, limitations of heterogeneous systems are pointed out in the form of a non-uniform programming environment, which limits the efficiency of the programming of the architecture. An unified compiler, linker and profiler would increase the programming efficiency.

3.1.3 ADRES

ADRES is a VLIW processor tightly coupled to a CGRA [38]. The functional units of ADRES support Single Instruction Multiple Data (SIMD) instructions to exploit the high data level parallelism typical of digital base-band signal processing. ADRES is utilized on the IMEC SDR platform for the implementation of the inner modem signal processing.

IMEC SDR platform is composed of a mix of ASIPs and ASICs, in order to meet high performance and flexibility. Each one of the IPs was designed in-house at IMEC. The communication between IPs is assisted by a Direct Memory Access (DMA) and takes place over an Advanced Microcontroller Bus Architecture (AMBA). The architecture is based on a template and therefore allows the integration of different number of processing nodes, ensuring a high scalability of the platform.

3.2 DSP-based architectures

Reconfigurable hardware solutions are generally tailored to a particular application domain. Moreover, the programming interface of CGRAs might introduce limitations to the software interface because of limited mapping flows. A more flexible way to implement the signal processing is through the utilization of DSP-based architecture. In fact, DSP-based architectures offer to programmers a familiar environment based on the Von-Neumann paradigm. Therefore, most digital base-band signal processing is done today on DSP cores. DSP cores have followed the same trend of processor cores moving towards multi-core implementations. The following subsections introduce and analyse some DSP-based solutions for software defined radios.

3.2.1 MuSIC platform

MuSIC platform is the SDR solution proposed by Infineon [45]. The digital base-band is centered on the utilization of DSP cores and accelerators to speed-up the execution of well structured and computationally intensive kernels, such as FFT and Finite Impulse Response (FIR) filters. The DSP is an optimization of a previously designed DSP core in order to match better the requirements of wireless communications. The control of the processing elements, both DSP and accelerators, is managed by a general purpose core.

3.2.2 GENEPY

GENEPY, developed by CEA and the University of Montpellier II, is a homogeneous multi-processor architecture targeted for wireless communications [28]. In particular, GENEPY was designed to improved the flexibility and the scalability of solutions

for SDRs, due to the limitations faced by previous research based on heterogeneous solutions [18, 32].

The tile of GENEPY architecture, named SMEP, is composed of two different parts: one for data management and one for data processing. Two DSP cores take care of the data processing. Each DSP can perform up to four 16-bit multiplications, running at 4000 MHz. For the data management part, a 32KB Random Access Memory (RAM) and four logical buffers are integrated in the tile. The buffers are dynamically reconfigurable to support different applications' requirements. The SMPE comes in two different versions, named respectively v0 and v1. The v0 SMEP interacts with the other tiles through a communication configuration controller. Therefore, the computing is homogeneous but the control of each tile is shared between the tile itself and a host processor. In v1, the communication configuration controller is replaced with a control processor, realizing a fully homogeneous architecture with fully distributed control. The distributed control allows a higher degree of flexibility, allowing eventually the implementation of future wireless standards.

3.2.3 SODA and Ardberg

SODA is a DSP based multi-processor architecture targeted to SDR applications [33]. SODA system consistes of a controller processor and four ultra wide SIMD (32-way 16-bit integer data path) processing elements, which provide the required processing power. The communication between the processing elements takes place over a shared bus. The communication is assisted by a DMA engine which acts as system interface between the controller and the SIMD. SODA is synthesised in 180 nm technology and operates at 400 MHz. Ardberg architecture was proposed in [53], as an evolution of SODA architecture. Ardberg introduces optimized SIMD processing elements, which are even wider than the ones utilized in SODA, in order to push the performance to the extreme. Ardberg was synthesised in 90 nm and operates at 350 MHz.

3.3 Architectural limitations

The trend to utilize multi-processor architectures for the implementation of the digital base-band in flexible radios is clear. Efficient MPSoC solutions have been proposed

both by academia and industry. However, when looking to future directions the high degree of flexibility and scalability become an indispensable requirement for the radio architecture. Powerful DSP cores and CGRAs offer today a good trade-off in terms of computational power and computational efficiency. However, a limiting factor for these architectures resides in the system scalability and complexity.

High degrees of flexibility and scalability can be achieved through homogeneous architectures. Because of the level of flexibility and scalability offered, growing importance is given today to homogeneous solutions. This applies not only to the wireless communications domain but also to other domains, such as multimedia and computer vision [14]. Homogeneous architectures offer system architects the possibility to obtain the required performance via the scaling of the number of computational nodes.

Beside hardware scalability, homogeneous systems have other appealing properties. Since the design is built around a single computational element, the design and verification costs are kept under control. Verification is today a very important step of the design flow and also one of the most expensive and time consuming phases [11]. Moreover, when realizing designs in UDSM technologies, different factors should be considered. Important issues in UDSM technology are the Process, Voltage, Temperature (PVT) variations. PVT variations can have a negative influence on the Integrated Circuit (IC) yield. Therefore, the design of smaller and simpler cores, which cover less silicon area, can mitigate the design issues arising from high PVT variations, typical of UDSM technology nodes [23].

Together with the hardware related issues, the software layer plays a big role in the design of future platforms. In fact, the software layer should be able to exploit the parallel resources made available by architectures. Moreover, the software should be scalable and easily adapt to the scaling of the HW platform. Therefore, new design and implementation approaches for the development of algorithms have to be investigated.

The following two chapters introduce a homogeneous multi-processor architecture based on a simple tile structure and algorithmic solutions to exploit the parallelism of the hardware. A simple tile structure allows a higher degree of integration of cores, enabling the implementation of a highly parallel platform. Moreover, the design and implementation of algorithms able to take advantage of the architectural parallelism

are introduced and evaluated.

4. NINESILICA ARCHITECTURE

In this chapter the reference platform, named Ninesilica, utilized in this research work is introduced; the architectural choices made for the design of the platform are underlined. In particular, details about the design and implementation of the platform, its FPGA prototyping and ASIC synthesis as well as an analysis of qualitative properties of Ninesilica architecture are discussed in detail.

Ninesilica architecture derives from the Silicon Café template [40], developed at the Department of Computer Systems [1] at Tampere University of Technology. The template allows the implementation of multi-processor architectures composed of an arbitrary number of nodes interconnected by a hierarchical NoC [5]. The type of processing elements is not specified in the template, which therefore allows the implementation of either homogeneous or heterogeneous multi-processor systems. Also, the implementation of clustered many-core architectures is possible, as discussed in more detail later in this chapter.

4.1 Hierarchical network-on-chip

The backbone of the NoC is a generic mesh topology, which can be tailored on the basis of different design parameters, such as number of nodes and their placement in the mesh. The composition of simple mesh structures as well as hypercubes is therefore possible. The hierarchical structure of the NoC is divided into two levels: the first level takes care of the communication inside the node, while the second level is responsible for the communication between nodes [4]. The integration of the two levels of communication allows the NoC to effectively implement different data delivery approaches: single-, multi- and broad-cast messages are supported.

[1] The department has been split and merged with the Department of Electronics and Communications Engineering and the Department of Pervasive Computing in January 2013. This work continues on the Electronics and Communications Engineering side.

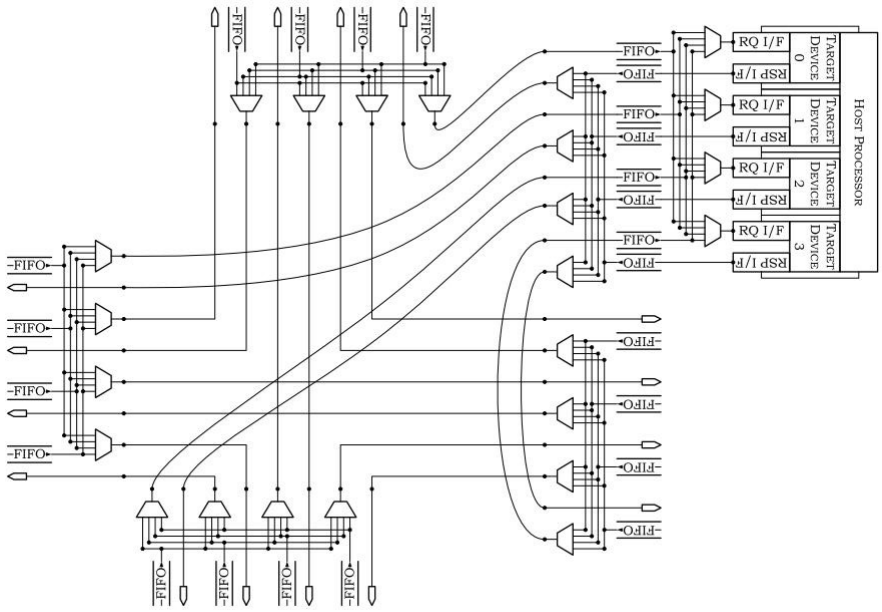


Fig. 4. *Detailed view of a single NoC routing node. [P3]*

A detailed schematic view of a single routing node of the NoC (also referred to as switch) is shown in Fig. 4. In the figure, only the interconnections are underlined, while the actual structure of arbiter/routers are stripped out to give a cleaner view of the routing options offered by each routing node. The switch has four ports for the interconnection with the neighbouring nodes and one extra port for the interconnection with the hosted computational node. Moreover, each one of the ports makes available 4 input queues which are multiplexed to the four other ports. Thus, each input to the port is differently queued in accordance with their destination. [6]

4.2 Ninesilica cluster

Ninesilica is the first instance of the Silicon Café template. As underlined by Fig. 5, Ninesilica is a homogeneous architecture composed of 9 processing nodes arranged into a 3x3 mesh topology. The choice of 9 processing nodes was driven by a trade-off between the number of processing elements, the processing power of the system, and the overhead introduced by inter-node communication and synchronization phases. A 3x3 mesh was an efficient trade-off between computational power and latency for

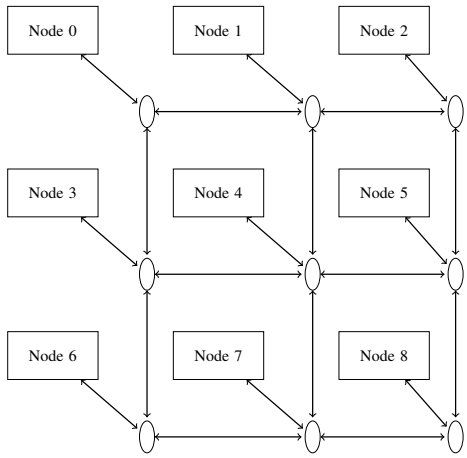


Fig. 5. *Schematic view of a Ninesilica cluster. [P3]*

the distribution of data and tasks among nodes. From a topological point of view, the node at the center of the mesh is in a particular position. In fact, the central node sees a uniform and balanced latency when accessing the surrounding nodes. Therefore, additional logical functionalities were given to the node in the central position. In particular, the central node takes care of the control of I/Os and the distribution of tasks and data, working de facto as the master node of the Ninesilica cluster.

A Ninesilica cluster can be utilized as an elementary building block for the implementation of clustered many-core systems. The clusterization of the architecture allows system designers and application developers to take full advantage of the high number of cores made available while retaining a high data locality and therefore a reduced communication overhead and workload across the whole NoC.

4.2.1 Computational node structure

The structure of a single computational node is shown in Fig. 6. Each node is composed of a hosted processor, data and instruction scratchpads and a network interface (NI). The NI takes care of routing the data to the global links of the network and at the same time provides the communication infrastructure for the node itself. For communication over the NoC, the NI utilizes a look-up-table (LUT), which contains 16 possible routing paths. The LUT is addressed by 4-bit of the data address whenever a remote write is requested. The routing paths can be redefined at run-time accord-

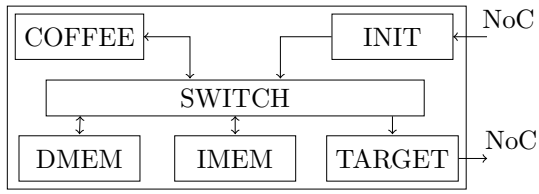


Fig. 6. *Schematic view of Ninesilica's computational node. [P3]*

ing to the application running on the platform. The choice of the processor core, for the purpose of this research work, is arbitrary and it does not affect the achieved results. The results are presented relative to architectures which utilize similar cores and therefore, switching the type of core will not modify the relative gains obtained but just the absolute performance. COFFEE RISC core [30, 31] was chosen as the computational element. The choice was due to the core's DSP-like features and to the free nature of the project.

4.2.2 Power management

Power consumption is one of the most crucial factors for the implementation of any modern system, and not just for hand-held devices. Therefore, an accurate power management technique has to be implemented to obtain high energy and power efficiencies. Two different approaches for the implementation of power and energy saving techniques were investigated for the Ninesilica cluster: 1) a software controlled architectural clock gating, allowing each computational node to switch into different functional profiles, and 2) a dynamic voltage and frequency scaling (DVFS) technique, to achieve higher levels of efficiency by tuning at run-time the power profile of each node of the architecture.

Architectural clock gating

Each node of the architecture was divided into 3 clock domains: one for the core, one for the memories and one for the network interface. Each clock domain can be independently switched on or off by the core itself or by a remote core through a software operation (a remote write to the clock gating controller). Each node can turn off its own clock domains whenever waiting for data to be processed and wake up

other nodes when data has been sent to those for further processing. The definition of three independent clock domains allows the utilization of different operating modes. In detail:

- *active mode*: all the clock domains are active and the node is fully functional. This mode is used when the node is actively processing data.

- *light sleep mode*: the only clock domain off is the one related to the core. This situation occurs, for example, whenever the node is not processing any data and a new set of data is streamed to the node. Thus, the memories and the NI need to be active to allow the streaming.

- *deep sleep mode*: core and memories are unclocked while the network interface is the only active domain inside the node. Such a working mode enables the node to save as much energy/power as possible while on hold for new data and new processing to be done. However, at the same time the node keeps the NI active allowing the other nodes to use it for re-routing of data in multi-cast scenarios.

- *off mode*: all of the clock domains are turned off and the node is in shut-down; hence, no activities can be performed. This mode is utilized whenever the node is not involved in any processing or data communication of the running algorithm.

The control of the nodes' mode is centralized and hosted by the master node. However, each node has the possibility to directly access the control register of the clock gating through the NoC. Each node is therefore able to put itself to sleep every time the computation has to stall (e.g. waiting of new data, tasks or synchronizations) and can switch other nodes that are targeted by its communications to a more active mode. The utilization of such a technique allows the switching of mode with the only latency being the access to the clock gating's controller. Because of the reduced latency, the nodes can always switch to a less active mode whenever possible, even if for just a few clock cycles.

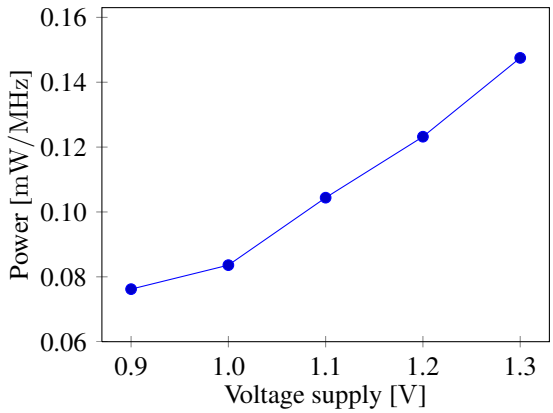


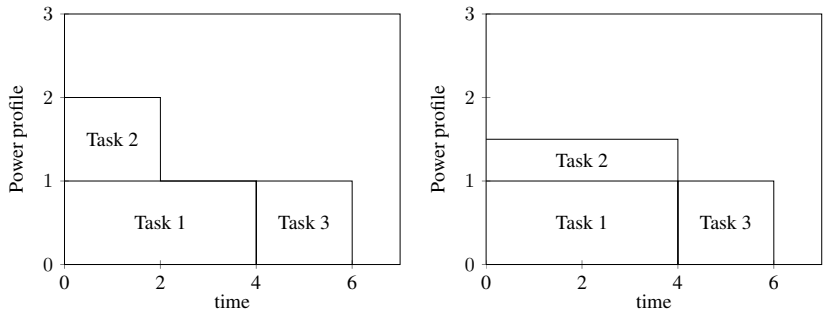
Fig. 7. *Trend example of power consumption as a function of voltage supply.*

Dynamic voltage and frequency scaling

The utilization of clock gating improves energy and power consumption, but it does not tackle static power consumption. For a larger impact on the power consumption, more complex approaches have to be considered. In particular, power consumption has a linear dependency from frequency and a quadratic dependency from voltage supply, as shown in the power characterization example of Fig. 7. An efficient system always runs at the lowest possible power profile in order to reduce its consumption to the minimum. For instance, Fig. 8 presents two examples of power allocations for a given set of tasks. In the examples, task 3 can be executed only after tasks 1 and 2 have completed; moreover, task 1 and task 2 are independent from each other and therefore they can be executed in parallel. In Fig. 8(a), a full power budget is allocated for the execution of task 2, which leads into the execution of the task and a stall phase (waiting for task 1 to be completed). However, if we can run task 2 on a lower power profile it is possible to obtain the same performance, but with lower power consumption (Fig. 8(b)).

Power gating

Other power management techniques could be deployed in order to tackle the static power consumption. One on these is power gating. Power gating allows the temporary switch-off of cores and therefore a complete elimination of the core's static power consumption during power-off time. However, due to the power-off a boot-up



(a) Tasks and power allocation 1 (b) Tasks and power allocation 2

Fig. 8. *Power profile allocation examples for DVFS-enabled systems.*

phase is required to restart the core when switching back to operating mode. The boot-up phase introduces a latency in the system response and for this reason, power gating might not be an efficient solution in the presence of high dynamics. Moreover, the physical implementation of power gating introduces some complications. In fact, retention cells or a level shifter are required in order to save the status of the core and memories at power off time. For these reasons, power gating was not investigated as a solution for the Ninesilica cluster.

4.2.3 Architecture's programming model

Ninesilica is fully programmable in ANSI-C and bases its software tool chain on the COFFEE RISC core compiler. In fact, each node of the architecture can be seen as an independent node with own scratchpad memories for instructions and data. The cores are cache-less and therefore at system level the integrity and coherence of data have to be maintained by the programmer at software level, with the utilization of synchronization semaphores.

The data exchanges and synchronisations between nodes are based on message passing. The NI of each node is mapped into a specific range of addresses in the memory space. An address that falls into such a range enables the NI and accesses the routing look-up-table in order to extract the routing information to the destination. The information of the routing is appended to the address and the packet is sent out on the network. Therefore, a data packet is formed of three parts: routing information, local destination address and data. To ensure a initial synchronisation between nodes and

Table 1. *Synthesis results of Quad-Ninesilica architecture on Altera Stratix IV (EP4GX530) FPGA device. [P3]*

Entity	**Resource Utilization**			
	ALUT	Logic Regs	DSP Blocks	%
COFFEE RISC	7,054	4,941	16	1.6%
NI	381	411	0	$< 0.1\%$
Central Node	7,617	5,308	16	1.8%
Processing Node	7,360	5,167	16	1.7%
Ninesilica	71,679	50,897	144	17%
NoC switch (average)	2,155	1,489	0 0	0.5%
whole NoC	99,026	69,223	0	23.4%
Quad-Ninesilica	352,970	247,099	576	83.5%

for exchanging data information a mailbox space (96 Bytes) is mapped for each node at the same memory address. This mailbox can be used at the initialisation phase to exchange information such as addresses of local variables, variable attributes, and synchronisation flags.

4.3 Hardware implementation

In this section, the FPGA prototyping of Ninesilica architecture is presented. Furthermore, the FPGA synthesis of a clustered many-core architecture based on Ninesilica cluster is presented to underline the high degree of hardware scalability provided by homogeneous platforms. The clustered architecture is composed of 4 Ninesilica clusters. The software performance of the clustered many-core architecture (Quad-Ninesilica) is then discussed in Chapter 5.

For the evaluation of power and energy consumption as well the impact of DVFS-aware algorithms implementations, ASIC hardware synthesis and power profiling of a single Ninesilica node are provided.

Table 2. *Power estimation for a single Ninesilica node. [P6]*

Voltage [V]	**Frequency [MHz]**	**Total Power [mW]**	**Dynamic Power [mW/MHz]**
0.9	100	7.75	0.0762
1.0	120	10.16	0.0836
1.1	150	15.83	0.1044
1.2	170	21.14	0.1232
1.3	200	32.5	0.1475

4.3.1 FPGA prototyping

Ninesilica cluster and Quad-Ninesilica were synthesized and mapped on an ALTERA Stratix IV FPGA device (EP4GX530). The synthesized architecture included the architectural clock gating for power and energy estimation. Table 1 gives a breakdown of the FPGA's resource utilization for the Quad-Ninesilica. Finally, the multi-cluster architecture runs at a frequency of 115 MHz in a slow mode synthesis. In a fast mode set-up the operating frequency rises to 180 MHz.

4.3.2 ASIC synthesis

To evaluate the performance of DVFS, the synthesis and power characterization of a Ninesilica node were carried out. The synthesis was done utilizing 65 nm standard cell libraries. A single node requires 63 eq. Kgates, covering an area of 0.44 mm^2 (0.13mm^2 std-cell and 0.31mm^2 memories) [7]. Power profiles, at different voltage supplies, are collected in Table 2. The power profiles were obtained through hardware synthesis of a single node and with a default level of switching activity.

4.4 Ninesilica qualitative evaluation

The evaluation and comparison of different system architectures is not a trivial task: different parameters (e.g. ASIC versus FPGA implementation and technology node utilized) play an important role in the evaluation of an architecture. Therefore, this section focuses on a qualitative evaluation of the Ninesilica approach, rather than a

quantitative comparison with other architectural solutions. The evaluation is based on the findings summarised in Section 3.3.

Ninesilica architecture is a homogeneous architecture, which can be also utilized as elementary building block for the realization of many-core architectures. The possibility of using Ninesilica as a building block for many-core architectures and the utilisation of a single type of processing element, allows a high degree of scalability in terms of hardware resources.

The processing node of Ninesilica architecture, differently from the DSP architecture introduced in Section 3.2, was designed in a minimalistic fashion: the processing element is a light core and not a muscular SIMD or VLIW machine. These architectural choices were made in order to realize a platform that was able to deliver the required computational power through the scaling of the number of cores, and not through the performance of a single processing node. In fact, the realization of small processing nodes allows a more successful implementation of the system in UDSM technologies, mitigating the issues related to PVT variations.

From the software point of view, a homogeneous solution based on the copy and paste of clusters and/or nodes allows the implementation of software in a modular way, leading into a high degree of scalability and performance.

5. DESIGN AND IMPLEMENTATION OF SDR ALGORITHMS ON NINESILICA

The design and implementation of SDR applications on the Ninesilica platform are presented and analysed in this chapter. In particular, the kernels of different communication systems, utilized as study cases, their implementation on the Ninesilica architecture and the achieved results are discussed in detail.

5.1 Algorithms

The implementation of algorithms on multi-processor architectures is generally not trivial [36]. In fact, the introduction of data communication overhead and synchronization barriers might mitigate the advantages that certain algorithm implementations have compared to others. Therefore, efficient algorithms on single core architectures cannot always be ported, with high parallelization levels, to multi-core architectures. On the other hand, a less efficient algorithm could benefit more from parallelization, leading into a more efficient mapping. The following subsections describe in detail significant kernels of major communication systems ported on Ninesilica and the implementation strategies adopted.

5.1.1 W-CDMA system

The Wideband Code Division Multiple Access (W-CDMA) system is at the base of 3G mobile communications. The technical characteristics of the W-CDMA can be found in [2]. The communication system is based on the utilization of spreading codes and the most computationally intensive kernels are related to the computation and evaluation of correlation values. Timing synchronization was chosen as a representative algorithm for the W-CDMA system.

Timing synchronization

The timing synchronization for W-CDMA systems can be seen as a composition of two distinctive kernels: the cell search algorithm [51] and multi-path delay estimation [22]. The main task of the cell search algorithm is to maintain synchronization between the mobile handset and the transmitting cell that offers the best Signal-to-Noise Ratio (SNR); the multi-path delay estimation algorithm takes care of the identification and the classification of multi-path components to align the receiver on the path offering the best SNR.

Both the cell search and the multi-path delay estimation algorithms are based on the same base kernel: computation and evaluation of correlation values. Independently from the algorithm deployed, the correlation kernel can be mathematically formulated as

$$Corr_n = \frac{1}{L} \cdot \sum_{i=0}^{L} R_{i+n} C_i^* \tag{1}$$

where $Corr_n$ is the $n-$th correlation value, R_{i+n} is the $(n+i)-$th sample of the incoming data stream, C_i and L are respectively the $i-$th sample and the length of the known sequence. Different parallelization strategies for Eq. (1) can be introduced; each one of the possible implementations introduces a different communication overhead and computational load to the nodes and therefore leads to different results. For instance, Eq. (1) could be rewritten as

$$Corr_n = \frac{1}{L} \cdot \sum_{j=0}^{N-1} \sum_{i=0}^{L/N} R_{i+\frac{L}{N} \cdot j+n} C^*_{i+\frac{L}{N} \cdot j} \tag{2}$$

where the computation of the correlation value has been divided into N identical parallel threads plus a sequential thread, which takes care of adding together the N partial results. Another possible approach can be to leave the computation of the correlation value as an atomic operation. Thus, each computational node would perform the following operation

$$Corr_{n \cdot N+M} = \frac{1}{L} \cdot \sum_{i=0}^{L} R_{i+n \cdot N+M} C_i^* \tag{3}$$

where M and N are respectively the thread identifier and the total number of computational nodes. The implementation of Eq. (2) and Eq. (3) requires a different dis-

Table 3. *Performance comparison between a single processor architecture, Ninesilica cluster and a Quad-Ninesilica architecture. [P3]*

Application	**Uniproc.** *(clk cycles)*	**Ninesilica** *(clk cycles)*	**Quad-Ninesilica** *(clk cycles)*	**Speed Up-1**	**Speed Up-2**
Corr. Point	12381	2546	688	18*X*	3,7*X*
Slot Synch. *(Fixed Part)*	52890764	7147387	1906649	27.7*X*	3.7*X*
Frame Synch.	3750593	471458	276428	13.6*X*	1.7*X*
Scramb. Code	149973	56203	56203	2.7*X*	1*X*
Cell Search	57410380	7802348	2366580	24.3*X*	3.3*X*

tribution of the data among the processing nodes, as well as different computational loads, leading to different performance.

To evaluate the scaling of the application as a function of the number of processing nodes, the whole cell search algorithm was initially mapped on a single Ninesilica cluster and then on a 4-cluster architecture (Quad-Ninesilica). Because of the clusterization it is possible to take advantage also of task parallelism, thus achieving higher speed-ups, as shown in Table 3.

5.1.2 OFDM system

The OFDM system has become one of the most utilized digital communication techniques for the implementation of the physical layer of wireless communications (e.g. IEEE 802.11/a/g/n and IEEE 802.16) due to its ability of providing high data rates. OFDM systems split the transmitted signal over different parallel sub-carriers, obtaining high data-rates. Furthermore, the utilization of orthogonal sub-carriers helps to mitigate issues such as multi-path channel fading and frequency selective fading. [24]

A typical OFDM transceiver structure is shown in Fig. 9. On the receiver side, which is the one requiring more computational power, the most critical kernels, from a computational point of view, are the demodulation and the timing synchronization. Demodulation is based on the computation of FFTs, where each FFT point represents an OFDM sub-carrier, whereas timing synchronization is based on the computation and evaluation of correlation values. The computation of correlation values is based on

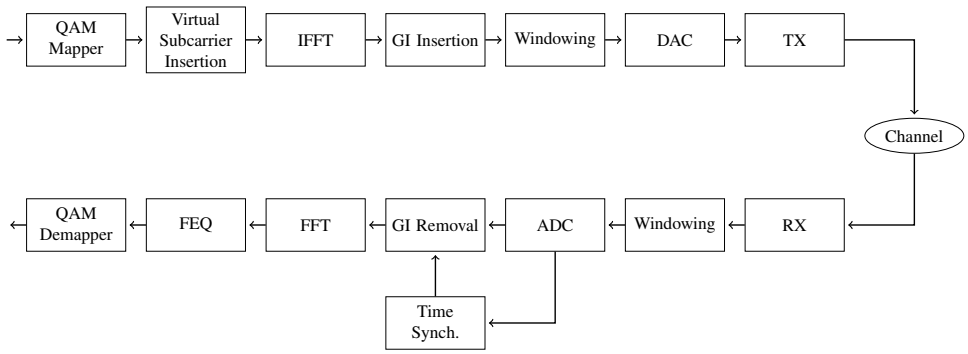


Fig. 9. *OFDM transceiver architecture. [P4]*

the same strategies introduced in Section 5.1.1 with a modification to the coefficients in order to match OFDM specifications for timing synchronization [8].

Demodulation: FFT algorithms

Many implementations of the FFT algorithm for OFDM system demodulation can be found in the literature. Each one of the proposed implementations targets a particular communication standard or a particular issue of the system design. Tang et al. proposed in [49] a multi-modal FFT processor for multi-standard Wireless Local Area Networks (WLAN) implementation aimed at reducing energy consumption and area; Hung et al. in [27] described a high performance FFT core for OFDM based communication systems able to perform variable size FFTs; the implementation of a non-power-of-two FFT on a reconfigurable architecture was presented in [54] by Wolkotte et al. A common point for the above mentioned works is the algorithm utilized for the implementation of the FFT.

The most common algorithm for FFT is the radix-2 algorithm proposed in [19]. Other radix-N algorithms have been developed based on the concept of radix-2. For example, radix-4 and radix-8 implementations ensure a simplification of the algorithm's complexity, reducing the total count of multiplications and additions [15]. As an example, Table 4 reports the total number of multiplications and additions for a 64-point FFT in the case of radix-$2/4/8$ algorithms. The implementation of these algorithms on single processor architectures follows quite well the trend of complexity reduction: a complexity reduction of 40% leads into a reduction of computation time of about 40%. However, when considering multi-processor architectures such dependability

Table 4. *64-point FFT algorithm complexity. [P7]*

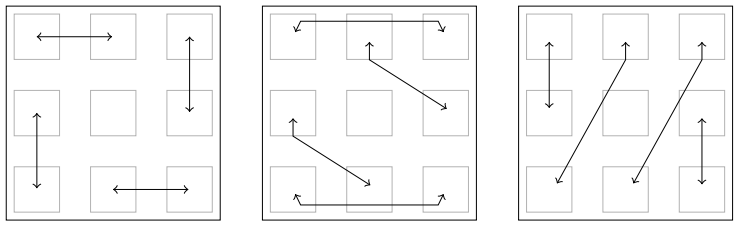
algorithm	Number of real MUL	Number of real ADD
Radix-2	264	1032
Radix-4	208	976
Radix-8	204	972

Table 5. *FFTs profiling on COFFEE RISC core and Ninesilica cluster. [P7]*

FFT size / Algorithm	# Clock cycles COFFEE core	# Clock cycles Ninesilica	Speed-up	Parallelization efficiency
64-point FFT				
Radix-2	22,214	3,773	5.9x	74%
Radix-4	10,937	3,388	3.2x	40%
Radix-8	10,282	3,880	2.6x	33%
2048-point FFT				
Radix-2	1,066,595	154,805	6.9x	86%
Radix-4	656,536	129,803	5x	63%
Radix-8	639,514	221,107	2.9x	36%

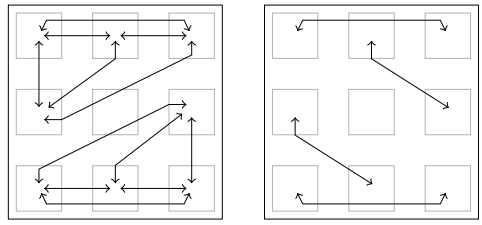
of the computation time as a function of the algorithm complexity might become weaker. In fact, data exchanges and synchronization phases between nodes play an important role in the performance of multi-processor systems.

The implementation of radix-$2/4/8$ on Ninesilica targeted the reduction of the total communication overhead due to data exchanges and synchronizations between nodes. In particular, the implementation strategy utilized was able to reduce the exchange phases required during the execution of the FFT to the minimum, independently from the FFT size [P2]. Fig. 10 shows the data exchange and actors involved in each communication step for the computation of the radix-2 algorithm. Due to the fixed scheduling of data exchange and actors the implementation of the algorithm could be based on a statical scheduling of tasks and data distribution. Radix-4 implementation reduces the number of data exchange phases required to two (see Fig. 11), but at the same time it increases the number of involved actors, leading to a more complex synchronization. Finally, the comparison between the implementation of radix-2/4/8 algorithms on Ninesilica and a single core architecture are collected in



(a) First data exchange. (b) Second data exchange. (c) Third data exchange.

Fig. 10. *Data exchange phases and actors in the implementation of the radix-2 algorithm on Ninesilica architecture. [P2]*



(a) First data exchange. (b) Second data exchange.

Fig. 11. *Data exchange phases and actors in the implementation of the radix-4 algorithm on Ninesilica architecture. [P2]*

Table 5. The analysis of the relative performance of each radix algorithm on the two reference architectures shows how the communication overhead of the multi-processor architecture can in certain cases mitigate the advantages of a reduced complexity algorithm. For example, the performance of radix-8 algorithm on a single core improves the computation time by a 50% factor when compared to the radix-2 implementation. However, on the multi-processor architecture the radix-8 degrades the absolute performance leading to an implementation result worse than radix-2. In fact, the synchronization and data exchange for radix-8 algorithm is performed in just one synchronization step, but it requires a data exchange and synchronization between all the nodes, introducing a large overhead of communication which nullifies the advantages of the radix-8 algorithm over radix-2.

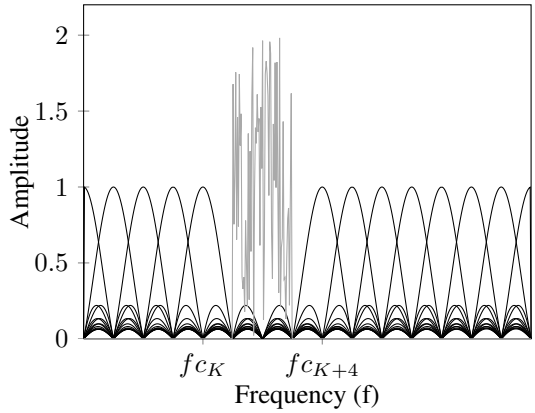


Fig. 12. *Example of sub-carriers' allocation for NC-OFDM systems. [P5]*

5.1.3 Non-contiguous OFDM system

OFDM offers a robust communication system and high data-rates, therefore it is highly exploited in the latest communication systems (e.g. IEEE 802.11a/g/n, WiMax and 3gpp-LTE). However, a higher degree of flexibility will be required by the next generation of wireless systems. For example, DSA communications require a higher level of flexibility when allocating the bandwidth for communication to take advantage, at run-time, of the actual available bandwidth. For this reason, variants of the OFDM system were proposed. Non-Contiguous OFDM (NC-OFDM) [44] and Discontinuous OFDM (D-OFDM) [43] are examples of OFDM-based communication systems which allow a higher degree of flexibility in the allocation of bandwidth and sub-carriers' mapping. In particular, in NC-OFDM systems data can be transmitted across disjointed frequency blocks to avoid interference with other communication systems active on a certain frequency band. An example of such a scenario is shown in Fig. 12, where a NC-OFDM sub-carriers' allocation (black plot) avoids interference with a primary user allocation (gray plot). Indeed, in the reported example, the sub-carriers at frequencies $fc_{\mathrm{k}+1}$ to $fc_{\mathrm{k}+3}$ are deactivated to avoid interference with the primary communication system.

The possibility of transmitting over a non-contiguous set of sub-carriers enables a more efficient utilization of the spectrum and a dynamic allocation of sub-carriers in accordance with the current spectrum workload. This higher degree of flexibility turns into a new level of flexibility of the transceiver architecture. In fact, the unused sub-carriers are forced to zero on the transmitter side and will be ignored on

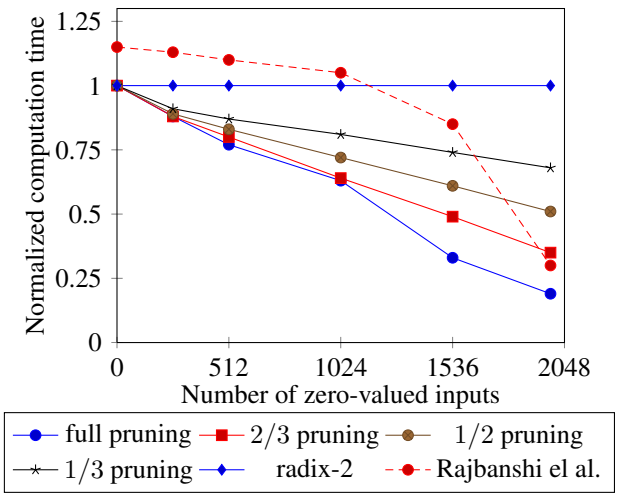


Fig. 13. *Normalized computation time of pruning algorithms for a 2048-point FFT. [P7]*

the receiver side. Thus, on the transceiver side simplifications can be introduced, making the computation of the kernels more efficient. For example, the introduction of pruning algorithms in the modulation/demodulation block can lead to a more computationally efficient and energy efficient implementation of the transceiver.

FFT pruning

The introduction of FFT pruning allows a reduction of algorithm complexity for the implementation of FFT kernels whenever zero values are fed as input. The reduction of complexity is obtained via the removal of dummy operations, such as multiplications involving 1 or 0 as multiplication factor or additions of zero terms. The theory behind FFT pruning is well known and has been studied extensively [9, 35, 48]. However, the implementation of FFT pruning on embedded systems is quite challenging due to implementation issues: most of the designed algorithms for FFT pruning do not consider carefully embedded systems' limitations (e.g. memory constraints). Moreover, the implementation of the control layer for the FFT pruning cannot be overly complex to take most advantage from the algorithm complexity reduction. For example, Zhang et al. in [56] proposed the implementation of an FFT pruning algorithm on a reconfigurable architecture, achieving a 30% reduction of the computation time when the input presents a sparseness level of 96%.

The elementary operation of the radix-2 algorithm is a butterfly, which produces two

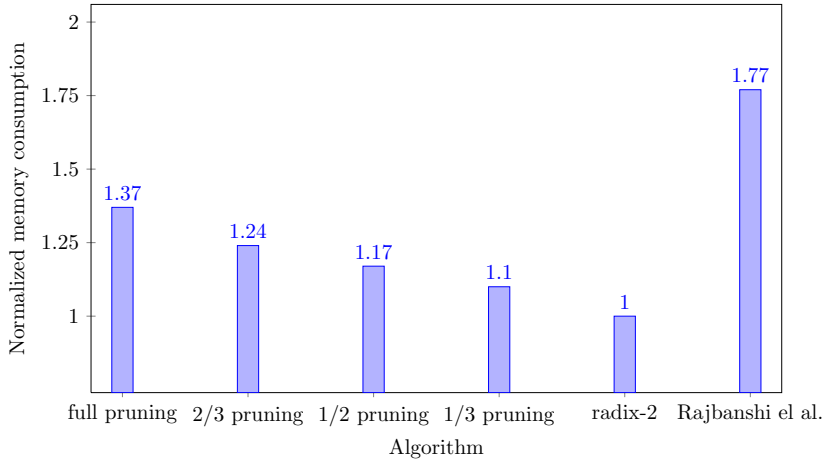


Fig. 14. *Normalized memory consumption of pruning algorithms for a 2048-point FFT. [P7]*

outputs of the two inputs. Therefore, there are four different scenarios (butterfly operations) when considering the possible zero distribution of the input. If both inputs are zero then the output will be set to zero (no operation); on the other hand, if both inputs are non-zero, a normal butterfly operation is performed; finally, if only one of the two inputs is zero valued the output will depend on the position of the zero term. This translated into a control plane for the FFT pruning algorithm might be costly. Therefore, simplifications have to be introduced. Rajbanshi et al. in [44] proposed an FFT pruning algorithm based on a configuration matrix, which avoids the utilization of *if-then-else* statements in the control plane, leading to better performance.

In [P7], a novel algorithm based on the Rajbanshi et al. configuration matrix is introduced. The algorithm is able to further reduce the memory requirements for the implementation of the pruning algorithm. Moreover, the proposed algorithm is able to achieve higher performance at lower pruning factors. The performance enhancements are mainly due to two factors: the simplification of the configuration matrix and the introduction of partial pruning. Their combination results in a more efficient and embedded system friendly implementation of the FFT pruning algorithm. Fig. 13 presents the normalized computation time of the proposed pruning algorithm (for a 2048-point FFT) and its partial pruning variants compared to the algorithm proposed in [44], both normalized to radix-2 performance. Finally, the memory requirements, normalized to the radix-2 algorithm are summarized in Fig. 14.

The FFT pruning algorithm is quite suitable for the evaluation of power and energy

Table 6. *Estimation of the energy and power saving through clock gating (CG) and DVFS techniques. [P6]*

Pruning profile	**Energy saving CG [%]**	**Energy saving DVFS [%]**	**Power saving CG [%]**	**Power saving DVFS [%]**
0	-	-	-	-
1	26.2	35.0	13.0	35.0
2	36.7	40.6	19.0	40.6
3	41.9	51.3	21.0	51.3
4	65.0	77.3	24.0	77.3
5	73.7	83.2	22.0	83.2
6	49.7	68.7	15	66.8
7	65.7	80.8	13	80.8
8	69.7	82.0	17	82.0
9	72.0	83.2	19	83.2

consumption since its computational complexity is data dependent. Therefore, an evaluation of the power saving achievable with clock gating and DVFS is more significant than the ones obtained analysing a statically loaded algorithm.

Table 6 summarizes the energy saving obtained utilizing the architectural clock gating introduced in Section 4.2.2, and the DVFS while performing the FFT pruning algorithm. Because of the data dependencies in the algorithm complexity, different scenarios were analysed. In particular, two different pruning distributions and different pruning levels were adopted. The first pruning scenario considers a contiguous pruning of the input, leading to an imbalanced workload on the computational node; the second scenario considers a uniform pruning distribution to obtain a balanced workload. The clock gating technique is able to efficiently tackle the energy consumption without having a large impact on the system performance. However, clock gating has a limited effect on the power consumption because it is not able to reduce static power consumption. On the other hand, the ability of DVFS to regulate the working point in terms of supply voltage and operating frequency gives high benefits in terms of power consumption, and therefore energy consumption.

6. SUMMARY OF PUBLICATIONS

In [P1], the implementation of the cell search algorithm for W-CDMA system on the Ninesilica platform is analysed. The Ninesilica platform is introduced in detail. Moreover, a power saving technique, based on a software controlled architectural clock gating, is also introduced. The achieved results show a high efficiency of the proposed partitioning of the algorithm as well as important energy saving via the utilization of the software controlled clock gating.

In [P2], different implementations of the FFT algorithm on the Ninesilica architecture are proposed and evaluated. The research work scope is to obtain a better understanding of how different implementations of the same algorithm scale in performance when moving from single-core to multi-core architectures. The achieved results underline how different implementations of the same algorithm change their behaviour when shifting from a single-core to multi-core platform. In particular, the overhead introduced by data exchange and synchronization phases between cores can mitigate, if not eliminate, the simplification introduced by certain implementations, while they can also emphasize other solutions.

In [P3], hardware and software scalability issues are analysed. The research work focuses on how the Ninesica platform behaves when scaling the number of nodes. A clustered many-core architecture, based on Ninesilica architecture, is proposed and the cell search algorithm for the W-CDMA system is ported on it. The implementation results show that Ninesilica can be efficiently utilized as a building block for the realization of many-core architectures, achieving high hardware and software scalability.

In [P4], the implementation of a generic OFDM transceiver on the Ninesilica platform is discussed in detail, utilizing as a case study the implementation of a receiver for the IEEE 802.11a/g standards. The analysis of the implementation highlights the pros and cons of a homogeneous multi-core architecture as a base-band engine for

wireless communications.

In [P5], the research focus is on the implementation of an efficient FFT pruning algorithm for cognitive radios. A novel FFT pruning algorithm is proposed and mapped on Ninesilica architecture. The algorithm leads into higher computational performance and lower memory requirements when compared to other algorithms. Moreover, the achieved results show that, in the context of cognitive radios, the proposed FFT pruning algorithm ported on the Ninesilica platform leads to a reduction in energy consumption in the order of 70%.

In [P6], the benefits of applying DVFS to a multi-processor system are highlighted. The main focus is on power and energy consumption. Moreover, the implementation detail of DVFS are analysed. The implementation of a data-driven algorithm with fixed real-time constraints is utilized as a case study. A comparison with the architectural clock gating introduced in [P1] shows that from an energy point of view both clock gating and DVFS achieve good performance. However, DVFS is also able to tackle power consumption, improving the hardware robustness to soft errors related to power integrity phenomena.

In [P7], the algorithm introduced in [P5] is discussed in greater detail. In particular, the research work focuses on the efficiency that can be achieved via FFT pruning without considering a specific mapping on a specific hardware platform. The obtained results show that the partial pruning approach is an effective trade-off between performance and the utilization of resources.

6.1 Author's contribution to the published work

The author of this thesis is the main contributor in all the publications collected in this research work. Each publication is the fruit of original work carried out by the author himself. The co-authors listed in each publication have contributed to the writing of the publications, given feedback about the research work, started discussions which led to an improvement of the achieved results as well as opened up new ideas, leading to future publications. Each one of the publications has been supervised by Prof. Jari Nurmi.

In [P1], the algorithm partitioning is the fruit of the author's work. The implementation of the clock gating technique controlled via software is due to a joint effort of

the author and co-authors. Open discussions about the software partitioning and the best possible way to reduce power and energy consumption led to the proposed clock gating scheme. [P2] was thought by the author as a way to better understand the influence of the intrinsic overheads of data exchange on multi-processor architectures. The partitioning and the study of the different implementations of the algorithm were the primary task of the author. In [P3], the author was responsible for the algorithm implementation on the Ninesilica architecture, the clustered many-core architecture synthesis on FPGA, as well as the management of the research effort that led to the published results. [P4] is a full effort of the author aiming to understand the pros and cons of the proposed platform when performing not just single algorithms of wireless communication systems but a full application; the whole application partitioning is based on previous publications and original ideas of the author. The author is also responsible for the entire mapping of the application on the Ninesilica platform and evaluation of the system performance. [P5], [P6] and [P7] are an effort of the author to understand better the potentialities of cognitive radios. In particular, [P7] introduces a novel algorithm for FFT pruning, while [P5] and [P6] focus on multi-processor architectures performing cognitive radio applications. Energy and power consumption were considered as key design parameters for the implementation and mapping of the algorithm on the Ninesilica architecture; both the ideas for the publications are the result of active discussions between the authors, co-authors and other colleagues. The author is responsible for the design and implementation of the proposed FFT partial pruning algorithm as well as the power consumption estimation and evaluation.

7. CONCLUSIONS

Multi-mode and multi-standard platforms are becoming the main design focus for the implementation of wireless communication systems. Flexible radios, implemented through the concept of software defined radios, are the answer provided by the research community. However, the utilization of the software defined radio approach introduces new challenges. At the digital base-band level, these challenges are related to the implementation of computationally powerful, flexible and power efficient platforms. Multi-processor architectures have been investigated as a feasible solution to meet such requirements.

Today's state of the art platforms, based on multi-processor architectures, provide a high computational power and system efficiency in terms of energy and computation. However, these systems provide limited scalability in terms of hardware and software. On the other hand, homogeneous multi-processor systems based on simple processor cores offer a higher degree of scalability, both in hardware and software. The high degree of scalability offered could become a potential advantage in the near future. In fact, a well established trend in computer architecture is to integrate more and more cores on the same chip, realizing massively parallel architecture. In this scenario, homogeneous clustered multi-processors could be a feasible design answer to these challenges.

From the application point of view, the design and implementation of algorithms has to be reviewed in order to exploit the high degree of parallelism made available by the platforms. Traditional algorithm designs might no longer be suitable for highly parallel architectures. Furthermore, the implementation of radio kernels in software allows the exploration of different and more flexible solutions, which were not possible through hardware implementations.

7.1 Main results

The main results achieved by this research work are twofold: 1) the definition of Ninesilica, a homogeneous multi-processor architecture as a representative platform for the implementation of software defined radios; and 2) the design and implementation of wireless communication algorithms able to take full advantage of the parallelism made available by Ninesilica architecture.

Ninesilica architecture is a homogeneous 3x3 mesh of computational nodes (CN). Each CN hosts a processor core as processing element. The central node in the architecture has the logical task of controlling the scheduling of tasks and data inside the cluster. The surrounding nodes can work on independent tasks as well as parallel accelerator. Dynamic power saving techniques were also implemented on Ninesilica with significant results in terms of power reduction. Moreover, Ninesilica cluster can also be utilized as an elementary building block for the implementation of clustered many-core architecture, ensuring a high scalability in terms of hardware.

The proposed algorithms for W-CDMA and OFDM based systems led to a high exploitation of the parallelism made available by Ninesilica architecture. The simulation results showed that the proposed implementations for the algorithms reached parallelization efficiency close to the theoretical limits. Moreover, the proposed algorithms together with the dynamic power management system provided by the Ninesilica platform were able to significantly reduce energy and power consumption.

The proposed algorithm implementations are not bounded to Ninesilica architecture and therefore could be ported to similar architectures. As an example, a homogeneous architecture based on a light SIMD processor core working at frequencies in the range of the GHz would offer the same relative performance as Ninesilica with an improvement of a factor of ten in absolute performance (e.g. replacing COFFEE core with a 2-way SIMD core at 800 MHz) while maintaining the mW/MHz figure in line with Ninesilica architecture. Such an architecture would be a competitive alternative to today's DSP core, providing the required performance of target applications as well as high power efficiency and system scalability in terms of hardware and software.

7.2 Open research issues

Many issues in the design of multi-processor architectures remain to be resolved. Programming languages and compilers able to efficiently port sequential code on top of a multi-processor architecture are still missing, even if a lot of effort and partial solutions have been proposed in the past decade. How data sharing and communication will be supported in the generation of thousand-of-cores has still to be clearly defined. In the specific case of Ninesilica architecture, the support of Message Passing Interface (MPI) or parallel programming languages like openCL could improve the programming efficiency of the architecture and further extend the portability of the software solutions proposed.

From the application point of view, more efficient communication systems will be developed, pushing radio platforms to become autonomous systems able to self-tune their communications in order to efficiently take advantage of the dynamic spectrum allocation. Therefore, more efficient algorithms and more efficient implementations are required. The algorithms should be able to exploit the provided parallelism without binding its implementation to a particular platform, enabling the portability of the solution across different radio platforms.

BIBLIOGRAPHY

[1] url = http://www.wirelessinnovation.org/ page visited on 8th May 2013

[2] url = http://www.umtsworld.com/technology/wcdma.htm page visited on 8th May 2013

[3] F. Agnelli, G. Albasini, I. Bietti, A. Gnudi, A. Lacaita, D. Manstretta, R. Rovatti, E. Sacchi, P. Savazzi, F. Svelto, E. Temporiti, S. Vitali, and R. Castello, "Wireless multi-standard terminals: system analysis and design of a reconfigurable RF front-end," *IEEE Circuits and Systems Magazine*, vol. 6, no. 1, pp. 38 – 59, 2006.

[4] T. Ahonen and J. Nurmi, "Programmable Switch for Shared Bus Replacement," in *Proc. of PhD Research in Microelectronics and Electronics PRIME 2006* , 2006, pp. 241–244.

[5] ——, "Hierarchically Heterogeneous Network-on-Chip," in *Proc. of International Conference on "Computer as a Tool" EUROCON*, 2007, pp. 2580–2586.

[6] T. Ahonen, "Network-based single-chip system architectures," Ph.D. dissertation, Tampere University of Technology, Tampere, FI, 2006.

[7] R. Airoldi, F. Garzia, and J. Nurmi, "Implementation of a 64-point FFT on a Multi-Processor System-on-Chip," in *Proc. of Ph.D. Research in Microelectronics and Electronics PRIME 2009*, 2009, pp. 20–23.

[8] ——, "Timing Synchronization for a Multi-Standard Receiver on a Multi-Processor System-on-Chip," in *Proc. of The 5th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC '10)*, 2010, pp. 153–155.

[9] R. G. Alves, P. L. Osorio, and M. N. S. Swamy, "General FFT pruning algorithm," in *Proc. of 43rd IEEE Midwest Symp. Circuits and Systems*, vol. 3, 2000, pp. 1192–1195.

[10] V. Baumgarte, G. Ehlers, F. May, A. Nckel, M. Vorbach, and M. Weinhardt, *The Journal of Supercomputing*, vol. 26, pp. 167–184, 2003.

[11] M. Fujita, I. Ghosh, and M- Prasad, Eds., *Verification techniques for system-level design*. Morgan Kaufmann, 2010.

[12] A. Beck Schneider, L. Lisboa, C. Arthur, and L. Carro, Eds., *Adaptable Embedded Systems*. Springer, 2013.

[13] F. Beltran, J. Gutierrez, and J. Melus, "Technology and market conditions toward a new competitive landscape in the wireless access market," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 46 –52, 2010.

[14] L. Benini, E. Flamand, D. Fuin, and D. Melpignano, "P2012: Building an ecosystem for a scalable, modular and high-efficiency embedded computing accelerator," in *Proc. of Design, Automation & Test in Europe Conference & Exhibition DATE'12*, 2012, pp. 983 –987.

[15] S. Bouguezel, M. Ahmad, and M. Swamy, "Improved radix-4 and radix-8 FFT algorithms," in *Proc. of International Symposium on Circuits and Systems ISCAS*, vol. 3, 2004, pp. 561–564.

[16] A. Chun, "Key lessons from the scalable communications core: a reconfigurable wireless baseband," *IEEE Communications Magazine*, vol. 48, no. 12, pp. 101 –109, 2010.

[17] A. Chun, K. McCanta, E. B. Sandoval, and K. Gulati, "Overview of the Scalable Communications Core: A Reconfigurable Wireless Baseband in 65nm CMOS," in *Proc. of IEEE Computer Society Annual Symposium on VLSI ISVLSI '09*, 2009, pp. 1–6.

[18] F. Clermidy, R. Lemaire, X. Popon, D. Ktenas, and Y. Thonnart, "An Open and Reconfigurable Platform for 4G Telecommunication: Concepts and Application," in *Proc. of The 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools DSD'09*, 2009, pp. 449 –456.

[19] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965.

[20] Federal Communication Commission, "Spectrum policy task force report," ET Docket No. 02135, Tech. Rep., 2002.

[21] F. Garzia, W. Hussain, and J. Nurmi, "CREMA: A coarse-grain reconfigurable array with mapping adaptiveness," in *Proc. of International. Conference on Field Programmable Logic and Applications FPL 2009*, 2009, pp. 708–712.

[22] E. Grayver, J. F. Frigon, A. M. Eltawil, A. Tarighat, K. Shoarinejad, A. Abbasfar, D. Cabric, and B. Daneshrad,, "Design and VLSI implementation for a WCDMA multipath searcher," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 889 – 902, 2005.

[23] K. K. Kim, and Y. B. Kim, "A Novel Adaptive Design Methodology for Minimum Leakage Power Considering PVT Variations on Nanoscale VLSI Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 4, pp. 517 – 528, 2009.

[24] J. Heiskala and J. Terry, *OFDM wireless LANs: A Theoretical and Practical Guide*. Sams, 2001.

[25] P. Heysters, G. Smit, and E. Molenkamp, "A Flexible and Energy-Efficient Coarse-Grained Reconfigurable Architecture for Mobile Systems," *The Journal of Supercomputing*, vol. 26, pp. 283–308, 2003.

[26] W. Huang, K. Rajamani, M. Stan, and K. Skadron, "Scaling with Design Constraints: Predicting the Future of Big Chips," *IEEE Micro Magazine*, vol. 31, no. 4, pp. 16 –29, 2011.

[27] C.-P. Hung, S.-G. Chen, and K.-L. Chen, "Design of an efficient variable-length FFT processor," in *Proc. of 2004 Internatioinal Symposium on Circuits and Systems, ISCAS*, vol. 2, 2004, pp. 833–836.

[28] C. Jalier, D. Lattard, A. A. Jerraya, G. Sassatelli, P. Benoit, and L. Torres, "Heterogeneous vs homogeneous MPSoC approaches for a mobile LTE modem," in *Proc. of the Conference on Design, Automation & Test in Europe & Exhibition DATE'10*, 2010, pp. 184–189.

[29] F. K. Jondral, "Software-Defined Radio–Basics and Evolution to Cognitive Radio," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 3, p. 652784, 2005.

[30] J. Kylliäinen, T. Ahonen, and J. Nurmi, *Processor Design: System-on-chip Computing for ASICs and FPGAs*. Springer, 2007, ch. General-Purpose Embedded Processor Cores: The COFFEE RISC Example, pp. 83–100.

[31] J. Kylliäinen, J. Nurmi, and M. Kuulusa, "COFFEE - a core for free," in *Proc. of International Symposium on System-on-Chip*, 2003, pp. 17–22.

[32] D. Lattard, E. Beigne, F. Clermidy, Y. Durand, R. Lemaire, P. Vivet, and F. Berens, "A Reconfigurable Baseband Platform Based on an Asynchronous Network-on-Chip," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 223 –235, 2008.

[33] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "SODA: A High-Performance DSP Architecture for Software-Defined Radio," *IEEE Micro Magazine*, vol. 27, no. 1, pp. 114–123, 2007.

[34] R. López, J. M. Camas Albar, M. Melchor, and D. Castells-Rufas, "How to obtain more powerful SDRs using multi-core architectures," in *Proc. of SDR forum 2009* , 2009.

[35] J. Markel, "FFT pruning," *IEEE Transactions on Audio Electroacoustics*, vol. 19, no. 4, pp. 305–311, 1971.

[36] G. Martin, "Overview of the MPSoC design challenge," in *Proc. of 43rd ACM/IEEE Design Automation Conference*, 2006, pp. 274–279.

[37] G. Masera, A. Baghdadi, F. Kienle, and M. C., "Flexible Radio Design: Trends and Challenges in Digital Baseband Implementation," *VLSI Design*, vol. 2012, 2012, article ID 549768.

[38] B. Mei, S. Vernalde, D. Verkest, H. Man, and R. Lauwereins, "ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix," in *Field Programmable Logic and Application*, ser. Lecture Notes in Computer Science, P. Cheung and G. A. Constantinides, Eds. Springer Berlin Heidelberg, 2003, vol. 2778, pp. 61–70.

[39] J. I. Mitola, and G. Q. J. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[40] J. Nurmi, "Silicon Café: a Heterogeneous Multi-Processor Platform based on Coffee (RISC Core)," in *Proc. of The 8th International Forum on Application-Specific Multi-Processor SoC*, 2008.

[41] A. Oates, "Reliability challenges for the continued scaling of IC technologies," in *Proc. of 2012 IEEE Custom Integrated Circuits Conference (CICC)*, 2012, pp. 1 –4.

[42] P. Paulin, "Programming challenges & solutions for multi-processor SoCs: An industrial perspective," in *Proc. of 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011, pp. 262 –267.

[43] J. D. Poston and W. D. Horne, "Discontiguous OFDM Considerations for Dynamic Spectrum Access in Idle TV Channels," in *Proc. of IEEE International Symposium on New Frontiers Dynamic Spectrum Access Networks*, vol. 1, 2005, pp. 607–610.

[44] R. Rajbanshi, A. M. Wyglinski, and G. J. Minden, "An Efficient Implementation of NC-OFDM Transceivers for Cognitive Radios," in *Proc. of 1st Int Cognitive Radio Oriented Wireless Networks and Communications Conference*, 2006, pp. 1–5.

[45] U. Ramacher, "Software-Defined Radio Prospects for Multistandard Mobile Phones," *IEEE Computer Magazine*, vol. 40, no. 10, pp. 62 –69, 2007.

[46] J. Reed, *Software Radio - A Modern Approach to Radio Engineering*. Prentice-Hall, 2002.

[47] W. Savage, J. Chilton, and R. Camposano, "IP reuse in the system on a chip era," in *Proc. of The 13th International Symposium on System Synthesis*, 2000, pp. 2 –7.

[48] D. Skinner, "Pruning the decimation in-time FFT algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 2, pp. 193–194, 1976.

[49] S.-N. Tang, C.-H. Liao, and T.-Y. Chang, "An Area- and Energy-Efficient Multimode FFT Processor for WPAN/WLAN/WMAN Systems," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 6, pp. 1419 –1435, 2012.

[50] C. H. K. van Berkel, "Multi-core for mobile phones," in *Proc. of the Conference on Design, Automation & Test in Europe & Exhibition DATE'09*, 2009, pp. 1260–1265.

[51] Y.-P. E. Wang and T. Ottosson, "Cell search in W-CDMA," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 8, pp. 1470–1482, 2000.

[52] Z. Wang and T. Arslan, "A low power reconfigurable heterogeneous architecture for a mobile SDR system," in *Proc. of 2004 Internatioinal Symposium on Circuits and Systems, ISCAS*, 2009, pp. 2025 –2028.

[53] M. Woh, Y. Lin, S. Seo, S. Mahlke, T. Mudge, C. Chakrabarti, R. Bruce, D. Kershaw, A. Reid, M. Wilder, and K. Flautner, "From SODA to scotch: The evolution of a wireless baseband processor," in *Proc. of The 41st IEEE/ACM International Symposium on Microarchitecture, 2008. MICRO-41*, 2008, pp. 152 –163.

[54] P. Wolkotte, M. van de Burgwal, and G. Smit, "Non-Power-of-Two FFTs: Exploring the Flexibility of the MONTIUM," in *Proc. of International Symposium on System-on-Chip*, 2006, pp. 1 –4.

[55] Q. Zhang, A. Kokkeler, and G. Smit, "A Reconfigurable Radio Architecture for Cognitive Radio in Emergency Networks," in *Proc. of The 9th European Conference on Wireless Technology*, 2006, pp. 35 –38.

[56] Q. Zhang, A. B. J. Kokkeler, and G. J. M. Smit, "An Efficient FFT For OFDM Based Cognitive Radio On A Reconfigurable Architecture," in *Proc. of IEEE International Conference on Communications ICC'07*, 2007, pp. 6522–6526.

[57] ——, "Cognitive Radio Design on an MPSoC Reconfigurable Platform," in *Proc. of 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications CrownCom 2007*, 2007, pp. 187–191.

[58] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

PUBLICATIONS

PUBLICATION 1

PUBLICATION 2

PUBLICATION 3

PUBLICATION 4

PUBLICATION 5

PUBLICATION 6

PUBLICATION 7

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3078-4
ISSN 1459-2045